Proposal of a function discovery system model using non-linear optimization method

Member Teruo Shimomura (Kyushu Institute of Technology) 
Non-Member B. B. Chhetri (Kyushu Institute of Technology) 
Member Senchi Serikawa (Kyushu Institute of Technology)

In this paper, we propose a new function discovery system model (S-Poly model) for the artificial life type discovery system (S-System), by incorporating non-linear optimization technique in it. This model uses Polytope method for optimizing the values of the constants of the function under discovery process while evaluating the fitness of it with the observation data. The new model significantly reduces the number of generation required for the discovery. Also, we propose two other models combining Polytope with Genetic Programming (GP-Poly model) and mutation (Mut-Poly model). All these models are found effective in the function discovery, while S-Poly model is found to be the fastest among the three.

Keywords: artificial life, genetic programming, function discovery, Polytope method

1. Introduction

J. Koza first proposed the function discovery system based on Genetic Programming (GP). However, the system has some disadvantages. We thus proposed a bug type of artificial life based system to overcome these problems. We called the system as S-System to indicate the incorporation of the concept of sexual and asexual reproduction in it. We then proposed a method to improve the search ability. However, the system still needed considerably long time for the discovery. Hence, in this paper, we propose a non-linear optimization method incorporated S-System that significantly reduces the time for the discovery.

2. Non-linear optimization method incorporated function discovery algorithm

GP and S-System are a kind of extension of Genetic Algorithm (GA) and uses the selection, crossover and mutation for function discovery. Table I shows how crossover and mutation change the function during the discovery process in GP and S-System. In both models, function type changes due to mutation. The difference between the two methods lies in the way of incorporating the crossover. In S-System, since the crossover is carried out only between the same type of functions, constants only change. Thus the schema does not get destroyed in S-System. This improves the local search ability. However, the time taken for the discovery is long.

Thus, S-System is characterized with the evolution of the same type of functions. Moreover, we can think that if the function type is limited to one, the constants in it can be optimized by using one of the non-linear optimization methods. From this, we incorporate a non-linear optimization method in S-System. We alter the previous S-System model in the following way.

(1) When the fitness is calculated, the constant \( K = k_1, k_2, \ldots \) in the gene (function \( f \)) is changed using Polytope method so as to minimize error. The new values of constant \( K' \) changed by the method is then replaced by \( K \).

Table 1 The change of function in GP and S-System.

<table>
<thead>
<tr>
<th>Model</th>
<th>crossover</th>
<th>Mutation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP</td>
<td>Function type(^{(1)})</td>
<td>Function type(^{(2)})</td>
</tr>
<tr>
<td>S-System</td>
<td>Values of constant only</td>
<td>Function type(^{(3)})</td>
</tr>
</tbody>
</table>

\(^{(1)}\)Includes the condition where the function type does not change.

(2) The algorithm for movement of bug is removed.
(3) Although the target of crossover is a slight change of constant \( dK \) in the original model, we change it to constant \( K \).

The other part of the algorithm of S-System is kept intact. Concerning non-linear optimization methods, we adopt Polytope method\(^{(4)}\). The reason is as follows: When there are many kinds of bugs, which have different type of functions, the evaluation of the gradient for all functions is impractical. Polytope method does not require the gradient function to be evaluated and hence we adopt this method.

The concept of movement, in which the value of constant change slightly, is not required in the new model, because the value of constant is changed by the Polytope method.

In general, the values obtained by Polytope method depend on initial values. In this model, the constant \( K \) in the gene (function \( f \)) corresponds to the initial values. This might results in a local minimum solution. In S-System, there exist many of functions with the different values of constant. This means that they have different initial values, so the probability of trapping in the local minimum is reduced. In addition, with the crossover in the above procedure (3), the values of constants (= initial values) can change greatly compared to the original model. This also helps to overcome local minimum trap.

3. Non-linear optimization method incorporated GP and mutation model

The new model is characterized with the change of the function type using mutation, with the optimization of the values of constants with Polytope method, and with the change of initial value using crossover. In the similar way,
the following methods can also be thought.  
(4) Remove (3) of above and make crossover between the different types of function. This is equivalent to combining GP model with the Polytope method. We call this as GP-Poly model.  
(5) Remove (3) and (4) and make algorithm without crossover. This can be thought as mutation combined with Polytope method. We call this as Mut-Poly model.  

4. Results  
We now present the results of using the proposed models for function discovery. The conditions with which S-Poly model was used for discovery are shown in Table 2. For GP-Poly model, the item 8, 10, and 13 were removed and for Mut-Poly model item 5, 6, 8, 10 and 13 were removed. We used all the three models to search the Kepler's law that was performed in the previous paper. All the models could discover the law at a few generations. Since the average of generations required by previous system to discover the law was about 300, we found the new models considerably faster. Computing time until discovery has been decreased to 1/20 or less. However, the difference in performance among the three models could not be understood due to an early discovery by all of them. Thus, we used the following expression to obtain dummy data sets for a complicated function type.  

\[ J = J_0 \cdot \left(1 - \frac{q \cdot V}{k_b \cdot T}\right)^{1}, \]  

where, \( J_0 = 1.0 \times 10^{-9}, \ q = 1.602 \times 10^{-19}, \ k_b = 1.382 \times 10^{-23} \), \( V \) was varied in steps of 0.01 from -0.05 to 0.1 in 16 steps and 3 value of \( T \), 243, 273 and 303 were used to evaluate \( J \). In total, 48 sets of data were generated as observation data. The three stated models were then used to search function \( J = f(V, T) \). 30 experimental runs were performed. The discovery rate was found to be 0.80, 0.77 and 0.77 for S-Poly, GP-Poly and Mut-Poly models. Fitness over 0.96 was used as the condition for discovery. In Fig. 1, the transition of fitness that is an average of 30 experimental runs are shown. From the figure, it can be seen that all the models are fast in discovery. Among the three models, S-Poly model is the fastest.  

Average computing times for each generation of S-Poly, GP-Poly and Mut-Poly models are 144, 180 and 115 seconds, respectively. Computing was performed with Pentium II 400 MHz processor operating on Windows NT system with C++ Builder 3.  

5. Conclusions  
In this paper, we proposed a new model called S-Poly model for function discovery by combining non-linear optimization method with the S-System. The proposed system could reduce the number of generations required for function discovery considerably. Also, by combining non-linear optimization method with genetic programming and mutation, we proposed GP-Poly and Mut-Poly models. All these models could discover the desired function effectively. The S-Poly model was found to be fastest on the average.  

We would like to thank Mr. Kazuhiro Morita of Kyushu Institute of Technology for his constructive comments. Asahi Technion Ltd has supported a part of this research.  

(Manuscript received July 13, '00)  

References  

Teruo Shimomura (Member) He received his B.S. and M.S. degrees from Kyushu Institute of Technology in 1969 and 1971, and his Dr. Eng. from Tokyo Institute of Technology in 1979.  
B. B. Chetri (Non-member) He received his B.S. from BUET, Bangladesh in 1993 and M.S. degree from Kyushu Institute of Technology in 2000 where he is now pursuing his PhD studies.  
Seiichi Serikawa (Member) He received his B.S. and M.S. degrees from Kumamoto University in 1984 and 1986, and his Dr. Eng. from Kyushu Institute of Technology in 1994.